ATSC Standard:
Video Watermark Emission

Doc. A/335:2022-03
31 March 2022

Advanced Television Systems Committee
1300 I Street, N.W., Suite 400E
Washington, D.C. 20005
202-872-9160

The Advanced Television Systems Committee, Inc. is an international, non-profit organization developing voluntary standards and recommended practices for broadcast television and multimedia data distribution. ATSC member organizations represent the broadcast, professional equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries. ATSC also develops implementation strategies and supports educational activities on ATSC standards. ATSC was formed in 1983 by the member organizations of the Joint Committee on Inter-society Coordination (JCIC): the Consumer Technology Association (CTA), the Institute of Electrical and Electronics Engineers (IEEE), the National Association of Broadcasters (NAB), the Internet & Television Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). For more information visit www.atsc.org.

Note: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. One or more patent holders have, however, filed a statement regarding the terms on which such patent holder(s) may be willing to grant a license under these rights to individuals or entities desiring to obtain such a license. Details may be obtained from the ATSC Secretary and the patent holder.

Implementers with feedback, comments, or potential bug reports relating to this document may contact ATSC at https://www.atsc.org/feedback/.

Revision History

Version	Date
Candidate Standard approved	30 November 2015
Standard approved	20 September 2016
Reference [2] updated to point to the published version of A/336:2017	27 February 2017
Amendment No. 1 approved	9 December 2019
Amendment No. 2 approved	2 February 2021
A/335:2022-03 published (references to ATSC documents updated)	31 March 2022

Table of Contents

1. SCOPE 5
1.1 Introduction and Background 5
1.2 Organization 5
2. REFERENCES 5
2.1 Normative References 5
2.2 Informative References 5
3. DEFINITION OF TERMS 6
3.1 Compliance Notation 6
3.2 Treatment of Syntactic Elements 6
3.2.1 Reserved Elements 6
3.3 Acronyms and Abbreviation 6
3.4 Terms 6
3.5 Extensibility 7
4. SYSTEM OVERVIEW 7
5. SPECIFICATION 11
5.1 Run-In Pattern 11
5.2 Watermark Data Symbols 11
5.3 Spatial Redundancy 12
5.4 1X Data Rate Video Watermark 12
5.5 2X Data Rate Video Watermark 13
ANNEX A : IMPLEMENTATION OF THE 1X WATERMARK DETECTOR 14
A.1 Overview 14
A.2 Description of Algorithm 14

Index of Figures and Tables

Figure 4.1 Marked video frame example. 8
Figure 4.2 Luma encoding – 2X System with 8-bit video. 9
Figure 4.3 Luma encoding – 1X System with 8-bit video. 10
Figure 5.1 Luma value calculation example. 12
Figure A.2.1 Example count of samples with a given value of luma. 14

Table 5.1 Horizontal Pixels per Symbol 12
Table 5.2 Luma Value Encodings for 1X System 13
Table 5.3 Luma Value Encodings for 2X System 13

ATSC Standard: Video Watermark Emission

1. SCOPE

This document specifies the emission format for video watermarks for use within ATSC 3.0 broadcasts.

1.1 Introduction and Background

The video watermark technology described in this document provides the capability to robustly embed ancillary data in the transmitted pixels of a video signal. It is intended to provide a data path for its ancillary data payload that can readily survive changes in video compression data rate, transcoding to other video compression codecs and delivery over legacy consumer HDMI interfaces. It is not intended to be a tamper-resistant or indelible watermark and it may be deliberately obliterated by an intermediary. Emission by a broadcaster of the video watermark is optional.

This video watermark emission technology is used to deliver the payload data described in ATSC A/336 [2].

1.2 Organization

This document is organized as follows:

- Section 1 – Scope and introduction.
- Section 2 – Lists normative references and informative documents.
- Section 3 – Provides a definition of terms, acronyms, and abbreviations for this document.
- Section 4 – System overview
- Section 5 – Specification of the video watermark emission format
- Annex A – Receiver guidelines for adapting to flexible modulation of the video watermark

2. REFERENCES

All referenced documents are subject to revision. Users of this Standard are cautioned that newer editions might or might not be compatible.

2.1 Normative References

The following documents, in whole or in part, as referenced in this document, contain specific provisions that are to be followed strictly in order to implement a provision of this Standard.

[1] IEEE: "Use of the International Systems of Units (SI): The Modern Metric System," Doc. SI 10, Institute of Electrical and Electronics Engineers, New York, NY

2.2 Informative References

The following documents contain information that may be helpful in applying this Standard.

[2] ATSC: "ATSC Standard: Content Recovery in Redistribution Scenarios," Doc. A/336:2022-03, Advanced Television Systems Committee, Washington, DC, 31 March 2022.

[3] SMPTE: "1920 x 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates," Doc. SMPTE ST 274:2008, Society of Motion Picture and Television Engineers, White Plains, NY, 29 January 2008.

3. DEFINITION OF TERMS

With respect to definition of terms, abbreviations, and units, the practice of the Institute of Electrical and Electronics Engineers (IEEE) as outlined in the Institute's published standards [1] shall be used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question will be described in Section 3.3 of this document.

3.1 Compliance Notation

This section defines compliance terms for use by this document:

shall – This word indicates specific provisions that are to be followed strictly (no deviation is permitted).

shall not – This phrase indicates specific provisions that are absolutely prohibited.

should – This word indicates that a certain course of action is preferred but not necessarily required.

should not – This phrase means a certain possibility or course of action is undesirable but not prohibited.

3.2 Treatment of Syntactic Elements

This document contains symbolic references to syntactic elements used in the audio, video, and transport coding subsystems. These references are typographically distinguished by the use of a different font (e.g., restricted), may contain the underscore character (e.g., sequence_end_code) and may consist of character strings that are not English words (e.g., dynrng).

3.2.1 Reserved Elements

One or more reserved bits, symbols, fields, or ranges of values (i.e., elements) may be present in this document. These are used primarily to enable adding new values to a syntactical structure without altering its syntax or causing a problem with backwards compatibility, but they also can be used for other reasons.

The ATSC default value for reserved bits is '1.' There is no default value for other reserved elements. Use of reserved elements except as defined in ATSC Standards or by an industry standards setting body is not permitted. See individual element semantics for mandatory settings and any additional use constraints. As currently-reserved elements may be assigned values and meanings in future versions of this Standard, receiving devices built to this version are expected to ignore all values appearing in currently-reserved elements to avoid possible future failure to function as intended.

3.3 Acronyms and Abbreviation

The following acronyms and abbreviations are used within this document.

ATSC – Advanced Television Systems Committee

HDMI – High-Definition Multimedia Interface

HDTV – High-Definition Television

PAM – Pulse-Amplitude Modulation

SMPTE – Society of Motion Picture and Television Engineers

3.4 Terms

The following terms are used within this document.

reserved – Set aside for future use by a Standard.

3.5 Extensibility

The video watermark emission format described in the present standard may coexist with, or be replaced by, future video watermark systems that employ different methods for embedding data into the video signal. Signaling of non-backwards-compatible watermark systems that use the first line of active video would employ a different value for the run-in pattern described in Section 5.1.

4. SYSTEM OVERVIEW

The video element of a broadcast program can encode a data stream that may be recovered from uncompressed video by the receiver. An ATSC 3.0 receiver that is receiving video via an HDMI interface can use this data stream for a variety of purposes, including hybrid (broadband) delivery of program elements such as those needed to support interactivity, dynamic ad replacement, service usage monitoring, and content identification.

The video watermarking technology specified herein involves modulation of the luma component of video within the top two lines of active video in each video frame. Two encoding options are offered, one providing a watermark payload of 30 bytes per video frame (a "1X" version), and the second "2X" version offering double that capacity.

The 1X version of the watermark encodes the payload data using luma values of black and a dark gray, which renders the watermark unobtrusive even if the display happens to present all 1080 lines of an HD image. The choice between larger payload and much-reduced visibility can be made by the broadcaster.

Figure 4.1 depicts one frame of marked video. The first portion of the top two lines of the 1080x1920 image are expanded to show the watermark. At the top, an example of the 2X version is shown and just below it, the 1X version. As shown, each group of 8 horizontal pixels represents an encoded data symbol. In the 2X version, each symbol represents two bits, while in the 1X version, each symbol represents one bit. In both cases, the lowest encoded value is encoded with the lowest (darkest) luma value. While in 8-bit video encoding, value 16 is specified as "black" in SMPTE ST 274 [3], certain values below black are specified for use in this application.

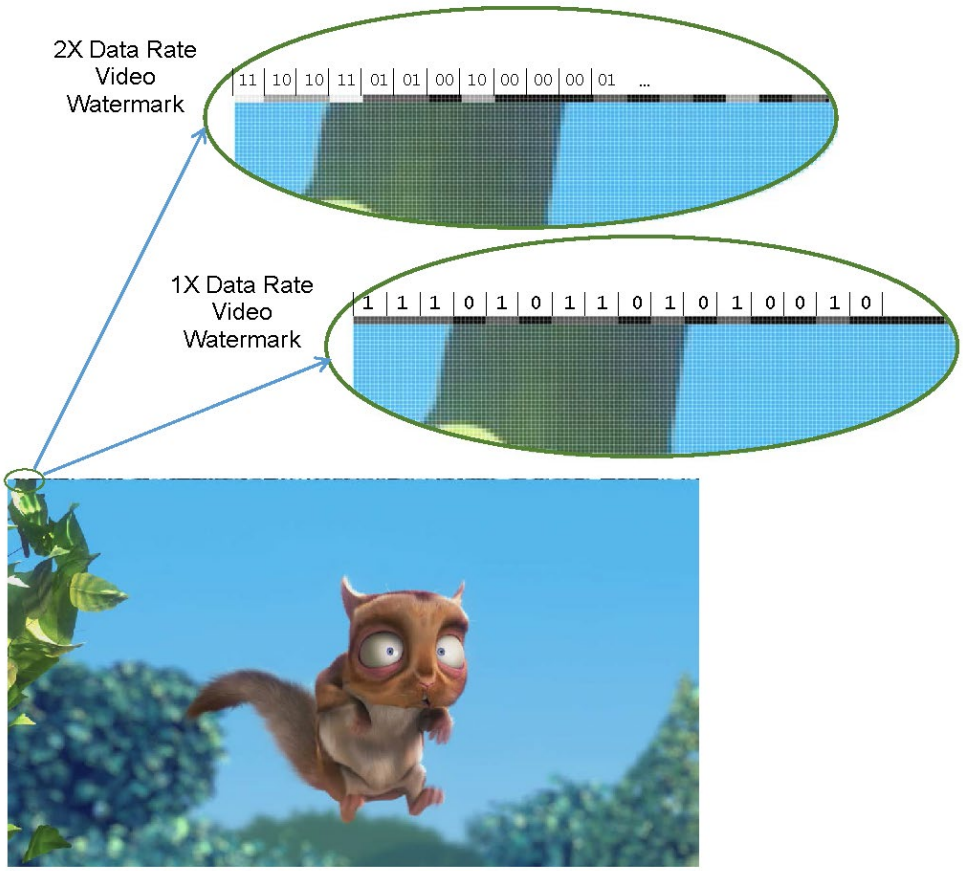

Figure 4.1 Marked video frame example.

Figure 4.2 shows the full range of luma values on the Y-axis for 8-bit video encoding and the range of black to white as defined in SMPTE ST 274 [3] of 16 to 235. As shown, for the 2X system, four levels of luma are used for the encoding, the black and white levels as well as two intermediate shades of gray (levels 89 and 162). Because the 2X system always employs the full luma range for encoding of the watermark, broadcasters are advised to use it only in distribution paths where receivers of the service are known to either have overscan activated or are capable of masking the watermark when present.

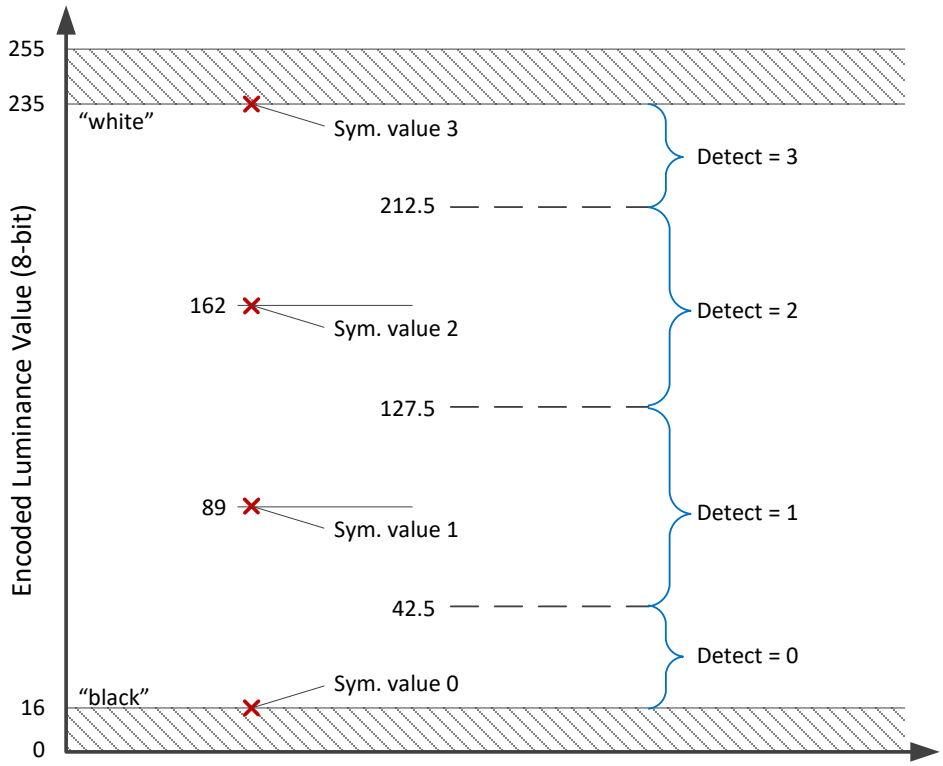

Figure 4.2 Luma encoding – 2X System with 8-bit video.

Modulation levels for the 1X system are flexible to allow the broadcaster to set the desired balance between visibility and robustness. The luma level for the "0" value of the symbol may be set to any value in the range 4 to 16 (for 8-bit video encoding), while the luma value used for the "1" value may be set to any value in the range 20 to 100.

The receiver is expected to take note of the modulation value in use and set a slicing level as appropriate. Figure 4.3 depicts two cases of this range. On the left, the modulation levels are 4 and 40, and the receiver sets an optimum slicing level of 22. On the right, the modulation levels are 4 and 100, and the receiver sets an optimum slicing level of 52. An algorithm that receivers may use to determine the optimum slicing level is given in Annex A.

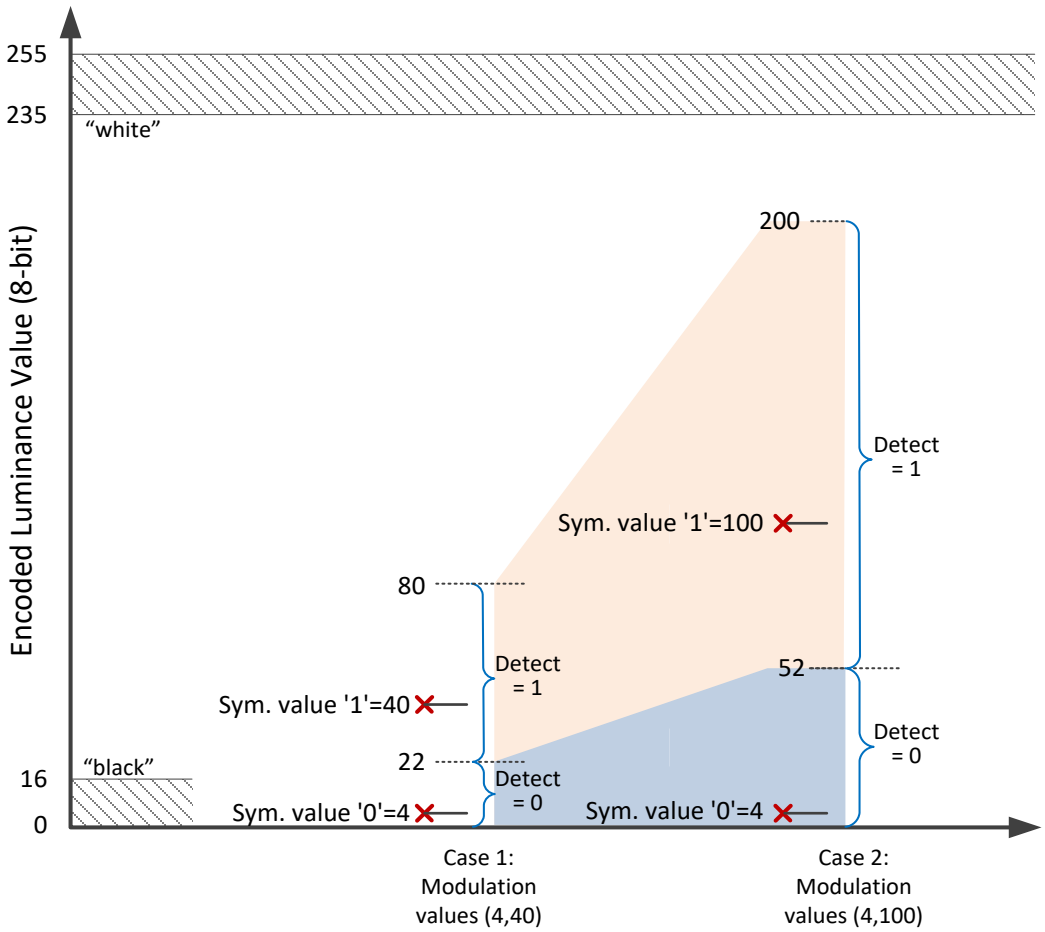

Figure 4.3 Luma encoding – 1X System with 8-bit video.

Selection of luma levels for the 1X system appropriate to a particular service environment is dependent on a number of factors, including: the number of viewers of the service whose receivers are expected to present the watermarked lines on-screen, the video resolution and formats used for the service, the video processing technologies (e.g., encoders and their associated bit-rates) in the distribution path, and the rate of watermark message data transmission required. Broadcasters are advised to take local market conditions into account in their selection of luma levels. In general, the greater the separation between luma values used to transmit the "0" and "1" symbol values, the more likely the message is to be detected after video processing present in the distribution path is applied. However, the higher the luma values of watermark symbols displayed to the viewer relative to program content, the more likely the watermark is to be visually obtrusive to viewers for whom it is presented on-screen. If on-screen display is expected to occur for a substantial number of viewers, broadcasters can minimize impact on perceived video quality by encoding the watermark at luma levels sufficiently low to preserve commercially acceptable subjective video quality and may need to transmit watermark messages with error correction coding and/or temporal redundancy to ensure reliable reception of the watermark data by receivers. Where necessary, watermark embedding implementations may adjust the luma levels employed based on

characteristics of the video content to optimize robustness while ensuring that a target video quality level is maintained.

5. SPECIFICATION

Digital data may be encoded within the luma component of the top two lines of active video. This section normatively specifies the emission format of the video watermark.

Two emission formats are specified: a normal and a high-rate version. The regular format, called the "1X Data Rate Video Watermark," or "1X system," encodes 30 bytes per frame of video, while the high-rate version, called the "2X Data Rate Video Watermark," or "2X system," doubles that to 60 bytes per frame.

The watermark payload is delivered within luma values. For marked content, the chroma values for video samples in lines 1 and 2 are not constrained by this specification and are expected to be ignored by watermark detectors.[1]

5.1 Run-In Pattern

For both the 1X and 2X systems, a run-in pattern consisting of 16 bits of encoded data is included within the first portion of the watermark payload. Receivers are expected to determine whether a given frame of video is marked or unmarked by first processing the luma values in the first portion of line one of uncompressed video to determine whether a valid run-in pattern is present. Receivers are expected to look for both the 1X and 2X run-in patterns to determine which encoding (if any) is in use in a given frame.

For both 1X and 2X systems the run-in pattern shall consist of a payload data value of **0xEB52** delivered most-significant bit first.

The receiver is expected to analyze the first line of active video and search for the appearance of this run-in pattern, modulated using either the 1X or 2X system parameters. If not found using the extraction algorithm suitable for the 1X system, it is expected to look for it using the 2X system.

5.2 Watermark Data Symbols

For the 1X system, two-level encoding is used so that each symbol represents one bit of payload data, while for the 2X system, four-level encoding is used and each symbol represents two bits of data. For both the 1X and 2X systems, 240 symbols shall be encoded across the full extent of the video line, regardless of the horizontal resolution of the video. Thus, for HD encodings of 1920 pixels horizontally, 8 pixels will convey the information of one symbol. For HD encodings of 1440 pixels, 6 pixels will encode one symbol. All symbols within a line shall be encoded across the

[1] Note that color system components can interact in unexpected ways as a result of clipping during conversions between color spaces (e.g., between YCbCr and RGB or vice versa). Implementers are advised to consider the potential impact of color space conversion when selecting values for the chroma channels of video samples that contain watermark payload data. It is further noted that such interactions can be avoided entirely if chroma channel values in the YCbCr color space are set to the midpoints of their ranges (i.e., 0x80 for 8-bit, 0x200 for 10-bit, and 0x800 for 12-bit video) or, in the RGB color space, if R, G, and B values are set to be equal to one another.

same number of pixels. Table 5.1 summarizes the number of pixels per symbol for typical horizontal resolutions.

Table 5.1 Horizontal Pixels per Symbol

Horizontal Resolution	Pixels per Symbol
1440	6
1920	8
3840	16

When the number of pixels of a video line divided by 240 does not result in an integer value (e.g., for video resolutions of 1280x720 or 2560x1440), the luma value of the pixel shared by two adjacent symbols shall be set to the weighted sum of the luma encoding values of the adjacent symbols, in proportion to the fractional spatial contribution of the symbol.

For example, for encodings of 1280 pixels, 5⅓ pixels will encode one symbol. Figure 5.1 illustrates the mapping of symbols to pixels and the resulting luma values for 32 pixels in the payload of a 1280-pixel-wide video frame, using 1X and modulation level 4 for symbol value "0" and 40 for symbol value '1'. In this example, the luma values for each pixel (abbreviated as "px" below) shared by two symbols are:

- px *i*+5: (⅓ * 40) + (⅔ * 40) = 40
- px *i*+10: (⅔ * 40) + (⅓ * 4) = 28
- px *i*+21: (⅓ * 40) + (⅔ * 4) = 16
- px *i*+26: (⅔ * 4) + (⅓ * 4) = 4

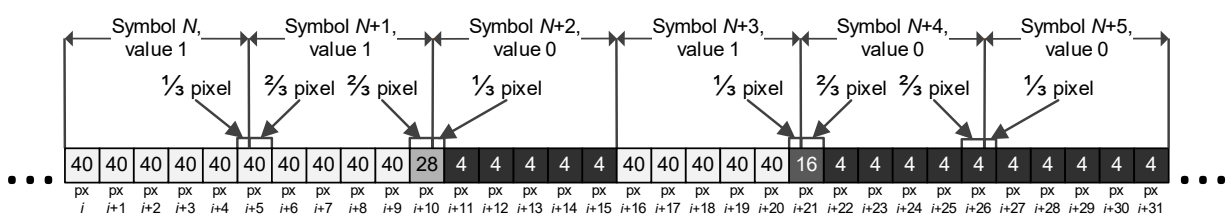

Figure 5.1 Luma value calculation example.

5.3 Spatial Redundancy

The watermark payload is recovered in the receiver by processing the first line of active video, however the encoder shall include the same watermark payload in the top two lines of active video in any given video frame. For interlaced video formats, this corresponds to the first active video line in each field. This spatial redundancy reduces the burden on the video encoder during the encoding process and helps ensure the watermark survives more aggressive compression.

5.4 1X Data Rate Video Watermark

Video signals encoded using the 1X version of the video watermark shall use 2-level modulation of the luma level to deliver one bit per symbol time. Luma values used to encode binary data in the 1X system watermark shall conform to Table 5.2 below. Values are shown for 8-, 10- and 12-bit video encoding. Luma values are shown in both hexadecimal and decimal format in the Table.

Table 5.2 Luma Value Encodings for 1X System

Bits per symbol	Encoded Data	Luma Value		
		8-bit	**10-bit**	**12-bit**
1	0	0x04 (4) to 0x10 (16)	0x010 (16) to 0x040 (64)	0x40 (64) to 0x100 (256)
	1	0x14 (20) to 0x64 (100)	0x050 (80) to 0x190 (400)	0x140 (320) to 0x640 (1600)
Minimum level difference		0x10 (16)	0x40 (64)	0x100 (256)

Note that in the 1X system a range of luma values is allowable for the "0" and "1" encoded data values. Higher luma values for the encoded data value "0" enable an increased range of overshoot protection in subsequent video processing and the ability to produce watermarks that remain fully within the normal range of luma specified by SMPTE ST 274 [3] at the cost of lower robustness against watermark decoding errors introduced by video compression or transcoding. Lower luma values for the encoded data value "1" result in less visibility at the cost of lower robustness against errors introduced by video compression or transcoding. Higher values can be used if greater robustness is desired. The receiver is expected to determine an appropriate slice point[2] for recovery of the watermark based on the observed luma values. To assist receivers in this, a minimum level difference is specified between the luma values used to encode data symbol values "0" and "1" within a video frame. Guidance for receiver manufacturers regarding how to determine the optimum slice point is given in Annex A.

5.5 2X Data Rate Video Watermark

Video signals encoded using the 2X version of the video watermark shall use 4-level modulation of the luma level to deliver two bits per symbol time. Luma values to encode binary data in the 2X system watermark shall conform to Table 5.3 below. Values are shown for 8-, 10- and 12-bit video encoding. Values are indicated in both hexadecimal and decimal format.

Table 5.3 Luma Value Encodings for 2X System

Bits per symbol	Encoded Data	Luma Value		
		8-bit	**10-bit**	**12-bit**
2	00	0x10 (16)	0x040 (64)	0x100 (256)
	01	0x59 (89)	0x164 (356)	0x590 (1424)
	10	0xA2 (162)	0x288 (648)	0xA29 (2592)
	11	0xEB (235)	0x3AC (940)	0xEB0 (3760)

[2] The slice point is the luma value used by the receiver to determine whether a received symbol represents a "1" or a "0." It would typically be set halfway between the luma value used to encode the "0" and the luma value used to encode the "1."

Annex A: Implementation of the 1X Watermark Detector

A.1 OVERVIEW

In the 1X system, the luma value usable to modulate the "0" value of the symbol is specified to lie within the range 4 to 16 and the "1" value of the symbol is specified to lie within the range 20 to 100 (8-bit video). For optimal recovery of the watermark, the receiver should set the slice point halfway between the value used to modulate "0" and the value used to modulate "1". This informative annex describes a simple algorithm a receiver can use to determine the best slice point when the 1X system is in use.

A.2 DESCRIPTION OF ALGORITHM

Given that in the 1X system the receiver does not initially know the optimal slice point to use to recover the data, the following algorithm can be used to find it. The algorithm is analogous to data recovery circuits in general use with analog communications systems: the first step in deriving the optimal slice point for 2-level PAM encoding is to find the location of the peak in the curve of frequency of occurrence of different luma values.

Figure A.2.1 illustrates an example analysis of watermark symbols. The graph was made by analyzing watermarks that were degraded by a process of video encoding/decoding. In all, 36,966 frames of video were analyzed, but similar results can be found with just a frame or two. The two levels of encoding used were 8-bit luma values 0 and 42.

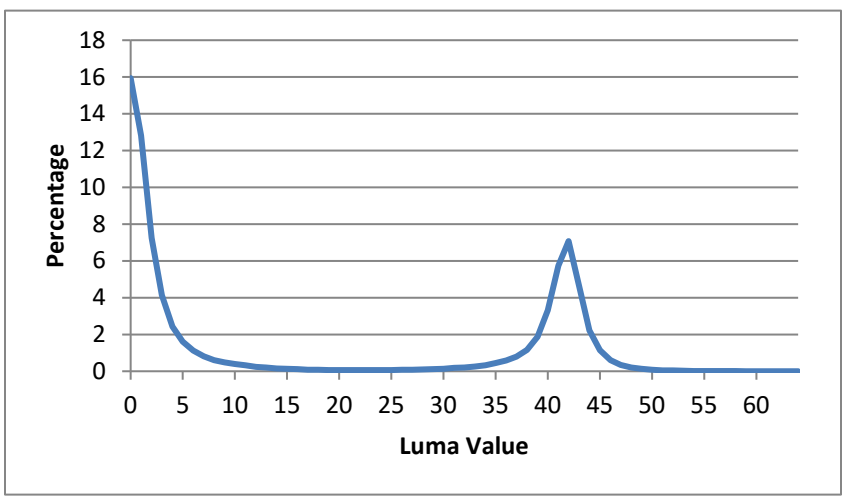

Figure A.2.1 Example count of samples with a given value of luma.

On the Y-axis, the graph plots the percentage of symbols that were found to be a particular value of luma given in the X-axis. For example, approximately 7% of all symbols had luma value 42 (the nominal, peak). Nearly zero had value 21 (exactly between the two luma values originally encoded).

A receiver can use an algorithm similar to the following to compute the peak value. Over a period of one or more frames:

1) For each frame, derive the symbol values by averaging each set of 8 pixels in the first line of active video and rounding to the nearest integer.
2) Count the number of symbol values occuring for each luma level in the range 20 to 100.
3) Determine the luma level having the highest number of observed values.

The following JavaScript code implements the above algorithm. Given these variables (data structures):

`rawLuma[]` – A 1920-element array containing the raw luma values (8-bit values)

`mySymbols[]` – A 240-element array containing the symbol values

`bins[]` – A 256-element array containing the accumulated number of instances where a symbol value matched the index value of the array. Example: `bins[30]` holds the total number of times a symbol value of 30 was found within some number of frames of marked video.

The following function derives the symbol values given the raw luma values:

```
function deriveSymbols() {  // derive symbol values by averaging luma over 8 pixels
        var i, j, a;
        for (i=0; i<240; i++) { // save average luma per symbol
                a = 0;
                for (j=0; j<8; j++) {
                        a += rawLuma[(i*8)+j];
                }
                mySymbols[i] = a/8;    // average luma of 8 sample
        }
}
```

The following JavaScript code could be used to derive the peak value of symbol luma over a range in which the upper encoded value may appear. For each frame, after the symbols are collected in mySymbols, the accumulation into bins can be done:

```
for (j=0; j<240; j++) {       // for each symbol
        n = Math.round(mySymbols[j]);
        bins[n]++;
}
```

Then, after one or more frames have been processed (bin data collected), the peak can be determined:

```
// find max value in range 21 to 100
m = 0;
peak = 0;
for (j=20; j<100; j++) {
        if (bins[j] > m) {
                m = bins[j];
                peak = j;
        }
}
```

This bin accumulation and peak detection process can be similarly applied within the luma range of 1 to 20 to establish an estimate for the encoding level for encoded data value “0”. For a 2-level encoding where the luma value used to encode “0” is estimated to be Z, the proper splice point would be (Z + `peak`)/2.

Testing has shown that even if only one frame is processed this way, a good first-approximation is given. Processing several frames can refine the value further.

— End of Document —